

Memoria del proyecto de manipulación de imágenes .pgm mediante C++

Alejandro Villanueva Prados Miguel Piñar Pérez
Francisco Bonillo González Gerardo Tirado García
Carlos Romero Cruz

Índice

1. Descripción General del Proyecto	2
1.1. Uso del programa	2
2. Descripción del trabajo de cada integrante	3
2.1. Francisco Bonillo González	3
2.2. Miguel Piñar Pérez	3
2.3. Alejandro Villanueva Prados	3
2.4. Gerardo Tirado García	4
2.5. Carlos Romero Cruz	4

1. Descripción General del Proyecto

Este proyecto implementa una manipulación de imágenes muy sencilla empleando para ello el formato .pgm y codificando la información de la imagen en una clase de C++. El programa nos permite manejar imágenes en blanco y negro con unas dimensiones máximas de 650x650 px. Utilizando el programa podemos Rotar una imagen, Blanquear la imagen, aumentar al máximo el contraste y sacar el negativo. Luego podemos leer imágenes desde ficheros .pgm y escribirlas con el mismo formato. Pasamos a describir más en detalle las funciones:

- Función miembro de entrada de datos. Se leen el identificador de archivo PGM (que es P2), el alto, el ancho, el valor maximo que puedo tomar el blanco (el 255). De estos valores solamente se almacena el ancho, alto y la propia imagen en forma de matriz.
- Función miembro de salida de datos. Va escribiendo sucesivamente el identificador, el valor del ancho, del alto, y el el valor de cada componente de la matriz que va en el rango de 0 a 255.
- Función miembro Blanquear. Pasa todos los valores de la matriz a 255, que es el valor máximo de luminosidad (el “blanco”).
- Función miembro Contraste Máximo. Si un valor de pixel de imagen esta por debajo de la mitad (127) este cambia a 0 (oscuridad máxima), en caso de que sea mayor o igual a la mitad, cambia a 255. Estos cambios se traducen en que todos los píxeles serán o bien blancos, o bien negros, dependiendo de su color original.
- Función miembro Negativo. Modifica el valor de cada píxel al valor del píxel menos 255, que es el valor asignado a blanco.
- Función miembro Rotar. Rota la imagen 90° hacia la derecha, esto se consigue primero realizando la traspuesta de la matriz y después se intercambian sucesivamente todas sus columnas, desde fuera hacia adentro.

1.1. Uso del programa

Para usar este programa, simplemente hay que añadir en la función ‘main’ de final.cpp las órdenes que quieras que realice el programa. Por defecto, rotará la imagen. Se puede sustituir por cualquiera de las funciones miembro de la clase (Consultar documentación usando Doxygen). Para utilizar el programa ejecutar en la terminal se deben redireccionar las entradas y salidas con ‘<’ y ‘>’, respectivamente.

2. Descripción del trabajo de cada integrante

2.1. Francisco Bonillo González

Mi parte del trabajo se basó principalmente en implementar una función miembro capaz de calcular el “negativo” de una imagen dada como entrada del programa.

Esto es, mediante dos bucles anidados según el alto y el ancho de una imagen, se va recorriendo cada componente de la matriz, que conforma cada uno de los píxeles de la imagen, y se va cambiando a su vez el número asignado que tenía (entre 0 y 255), por su complementario.

Por ejemplo, en el caso de que el píxel tuviera un valor de x , se calcularía pues el negativo – su complementario – que es $255 - x$. Ese x es el componente `pixel[i][j]` de la matriz de la imagen. Así se varía y se invierte la luminosidad de manera que la imagen queda en negativo.

Además, mi implicación en el trabajo ha sido de ayudante tanto en la sección de elaboración de memoria como en el desarrollo de otras funciones más generales.

2.2. Miguel Piñar Pérez

Mi trabajo consistió en:

- El struct inicial que almacenaba el ancho, el alto, y una matriz de 650x650 que almacenaba los valores de los píxeles de la imagen.
- La función de blanqueo. Esta función recorre toda la matriz cambiando el valor de todos los píxeles de la matriz a 255 (blanco).
- La función para trasponer una matriz, primera parte de la función de rotación. Para rotar una matriz 90° a la derecha hay que trasponerla y después cambiar sucesivamente sus columnas: la primera por la última, la segunda por la penúltima. . . La función de trasposición recorre una mitad de la matriz intercambiando los valores (i,j) por los (j,i) .

2.3. Alejandro Villanueva Prados

En este proyecto, mi trabajo ha consistido en gestionar el control de versiones, haciendo uso de la plataforma GitHub. Así todos los miembros hemos dispuesto de un lugar común para subir el código en el que trabajamos. Aparte también he redactado esta memoria, organizando las memorias del resto de los integrantes, añadiendo mi parte, y dando un formato a todo

usando L^AT_EX.

Con respecto a mi aportación al código, he implementado la función de entrada de datos, que lee desde un fichero los datos que indentifican a una imagen y que permiten manejarla por el programa, y también convertí el código de una versión anterior, basada en un struct, a una clase con diversos métodos, ya que así incluimos los últimos conocimientos adquiridos en la asignatura y también simplifica el código. Luego realicé la función ‘Swap’ que intercambia dos columnas dadas de la matriz. Por último me encargué de documentar el proyecto usando Doxygen.

2.4. Gerardo Tirado García

Por mi parte, me he encargado de implementar la función ‘Contraste’. La función Contraste tiene como elemento principal una matriz llamada píxel que pertenece a la clase PGM. Dentro de la función se han creado dos variables i, j. La función consiste en dos for anidados con dos if. Gracias a los for se accede a cada elemento de la matriz y comprobamos si ese elemento es menor que 127 o mayor o igual que 127. Si el elemento es menor que 127, se le asignará el cero, y si es mayor o igual, se le asignará el 255. Por lo tanto, obtendremos una matriz en la cual sus elementos son 0 o son 255. Así se aumenta el contraste a su máximo valor.

2.5. Carlos Romero Cruz

La parte que realicé en este proyecto fue la de diseñar la función de salida de datos e implementarla en el programa de manera que se crease correctamente el archivo PGM.

A la hora de iniciar la función de salida de datos, primero se escribe en el archivo la cabecera formada por la cadena mágica, el ancho y el alto de la futura imagen y el máximo valor de gris, y a continuación la sucesión de números que componen el archivo en forma de matriz con sus respectivos valores después de haber realizado las operaciones requeridas. La función realiza la salida mediante dos bucles for anidados. El primero recorre las filas del archivo PGM mientras que el segundo recorre las columnas. En cuanto ya se han escrito los números de una fila, se inserta un salto de línea para darle el formato de matriz requerido.